PPE Attitude Profile Reference for the Earth Orbit Raising Phase

Brad Humphreys[1]

Christine Schmid[1]

Michael Martini[1]

Melissa McGuire[1]

Kurt Hack[1]

Steven McCarty[1]

Mathew Holub[2]

Nino Tarantino[2]

Blake Leiker[2]

February 7, 2022

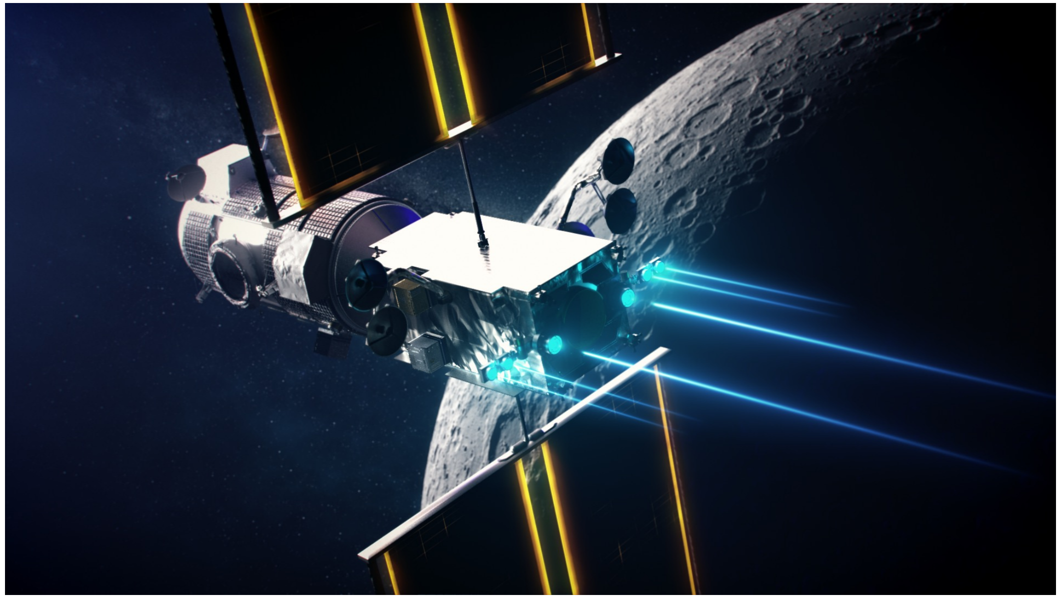
[1]NASA Glenn Research Center

[2]CACI NSS Inc

American Astronautical Society Guidance, Navigation and Control (GN&C) Conference

Background

- **First Components of NASA's Gateway**
  - Co-Manifested Vehicle = Power & Propulsion Element + Habitation and Lunar Outpost
  - CMV = PPE + HALO
- **~1 year Spiral Out to an Orbit around the Earth-Moon L2**
  - Earth Orbit Raising (EOR)
- **Relevant Spacecraft Features**
  - 8 Reaction Wheels + 24 RCS Jets
  - 7 Solar Electric Thrusters (6 can gimbal)
  - Rotating Roll out solar arrays



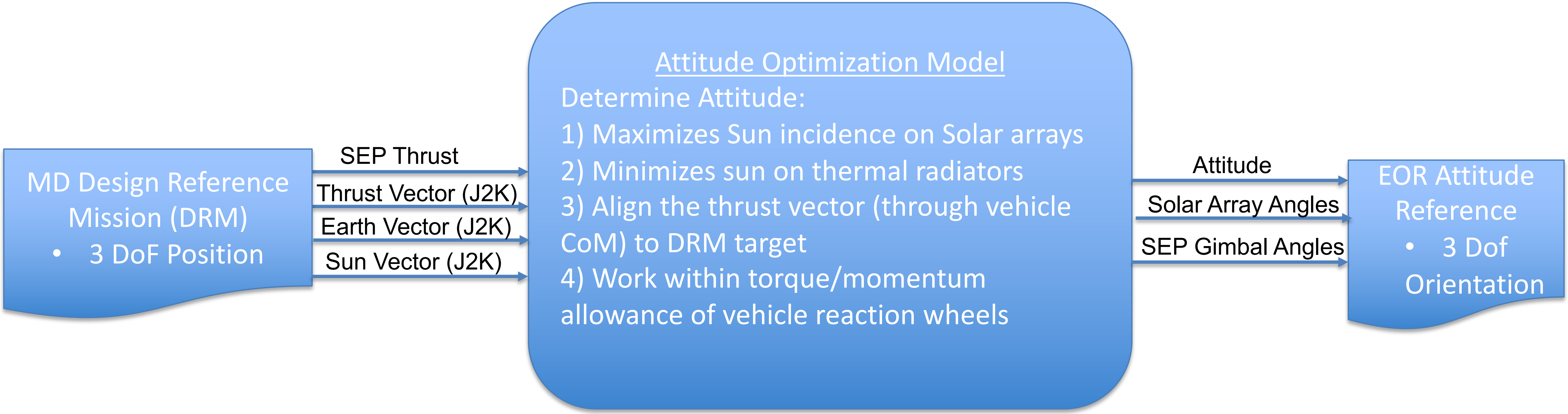
Background

- **Rationale for work**
  - Address attitude inquires from CMV stakeholders
  - Utilized to understand vehicle performance and attitude trade space
    - Reference for feedback from subsystems for flushing out attitude constraints
    - Currently focused on thermal/power/mission top level objectives. Jumping off point for discussions with Comm and other systems

- **Simulation is not a controls based model**
  - PPE GN&C has a high-fidelity controls-based simulation that implements control algorithms. This simulation is capable of both using NASA and Maxar (space craft prime) provided submodels

- **Simulation is an optimization-based model**
  - Generates the attitude within hardware performance constraints, but without detailed control algorithms

Model Description



- **Design Reference Mission (DRM)**
  - Mission Design Team develops a 3 DOF EOR trajectory
    - Uses ground rules and assumptions to investigate performance trades and ability to achieve mission goals
    - Selected reference mission identified
- **EOR Attitude Reference**
  - Compendium to DRM that adds Attitude data

Assumptions and Limitations

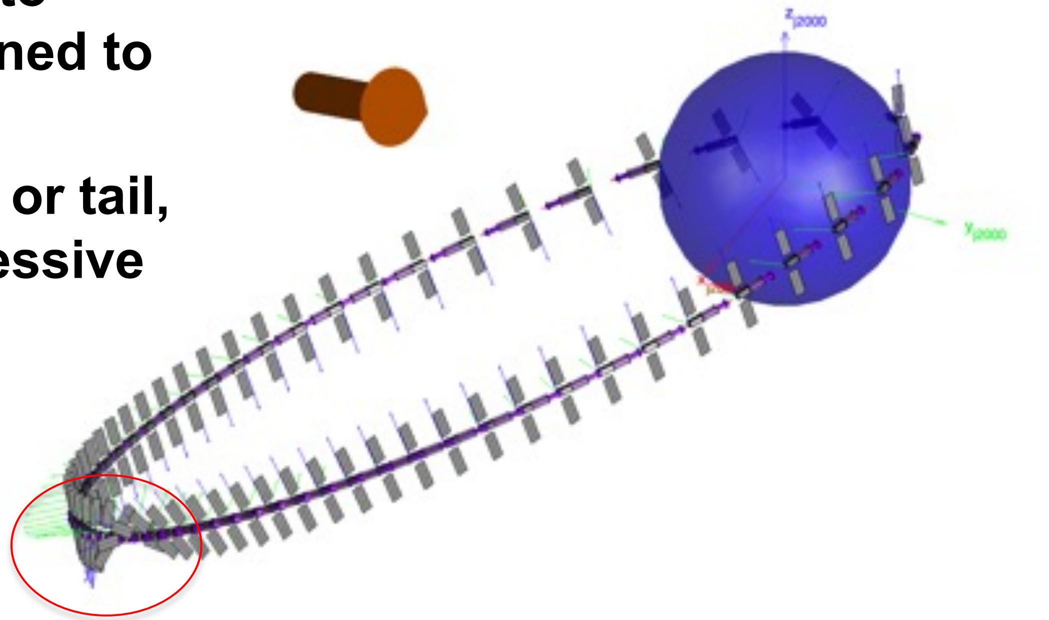
- **Fixed CoM and Base Vehicle Inertia Properties**
  - Location of solar array CoM is assumed fixed and on rotation axis
  - Effect of solar array rotation is considered

$$\mathbf{I_{veh}}(\theta_{sada,1}, \theta_{sada,2}) = \mathbf{I_{veh,base}} + \mathbf{I_{SA}}(\theta_{sada,1}, \theta_{sada,2})$$

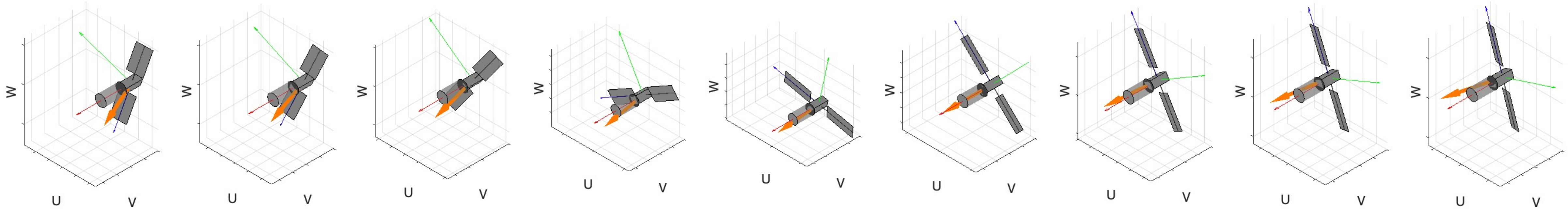
- **No SEP engine throttling**
- **Thrust misalignment does not cause appreciable change in DRM**
- **No Drag or Solar Radiation Pressure ... yet**

Key Hole Roll

- **So why can't we just continuously roll the vehicle to keep the sun exactly normal to the arrays and aligned to DRM thrust target?**
- **When the sun gets in close alignment to the nose or tail, a keyhole/snap-roll has to occur and requires excessive momentum usage**

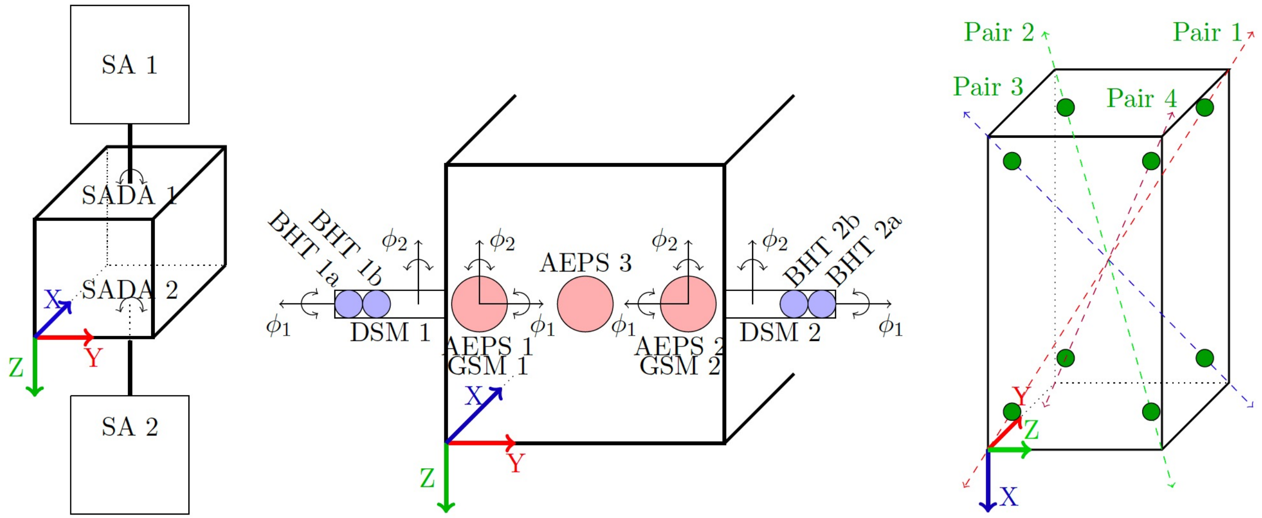


- *Sun vector shown in orange. Note perpendicularity to array axis.*
- *As sun vector (almost) aligns with X_{GACS}, vehicle is forced to roll*



Defining Known and Unknowns

- **Performing Optimization in the Direct Collocation framework**
  - Calculating at N number of nodes along trajectory
- **Unknown Trajectories (15)**
  - Reaction Wheel Torques (RW operate in pairs) (4)
  - Solar Array Drive Motor Torques (2)
  - Engine Gimbal Angles
    - AEPS 1, Angle 1 and 2
    - AEPS 2, Angle 1 and 2
    - BHT1a and BHT1a Engine, Angle 1 and 2
    - BHT2a and BHT2a Engine, Angle 1 and 2



- **Known Trajectories (16)**
  - Engine Thrust
  - Thrust Vector in J2000 Frame (x,y,z unit)
  - Sun to PPE Vector in J2000 Frame (x,y,z unit)
  - Earth to PPE Vector in J2000 (x,y,z unit)

Constraints

- **With the following constraints**
  - Functional
    - Limit the **torque of the reaction wheel motors** (not on speed, more about this later) (4)
    - Limit the **torque of the solar array drive motors** (2)
    - Limit the **extents of the gimbal angles** (8)
  - Mathematical
    - Initial conditions (18)
    - Dynamic consistency of equations of motion (18)
    - Unit normal constraint on the attitude quaternion (1)

Objectives

- **Objective Function to minimize**
  - Reaction Wheel minimum torque effort
    - Sum square of 4 RW divided by Max Torque Square to normalize ~[0:1]

$$J_{rw} = \frac{1}{4T_{rw,max}^2}(\mathbf{x}_{rw1}^T\mathbf{x}_{rw1} + \mathbf{x}_{rw2}^T\mathbf{x}_{rw2} + \mathbf{x}_{rw3}^T\mathbf{x}_{rw3} + \mathbf{x}_{rw4}^T\mathbf{x}_{rw4})$$

  - Maximize Sun Incidence on Arrays
    - Defining solar array incidence as:

$$\phi_n = ((\hat{\boldsymbol{r}}_{sada} \cdot \hat{\boldsymbol{r}}_{ppe,sun}^{Body})^2)^{1/2}$$

1: Aligned
0: Perpendicular
-1:Negative

    - Square and Root implements a continuous absolute value: either side of array can be active
    - For 2 arrays and moving to domain [0:1]

$$J_{sa} = \frac{1}{2N}\left(\sum_{n=1}^{N}(1-\phi_{sada1}),n + \sum_{n=1}^{N}(1-\phi_{sada2}),n\right)$$

  - Maximize alignment of vehicle thrust vector (through CoM) to target from DRM

$$\psi_n = \hat{\boldsymbol{r}}_{thrust} \cdot \hat{\boldsymbol{F}}_{ppe,thrust}^{Body}$$

$$J_{sep} = \frac{1}{N}\sum_{n=1}^{N}(1-\psi_n)$$

Objectives

- **Total Objective Function is weighted sum of the reaction wheel, solar incidence, and thrust alignment**

$$J = w_{rw}J_{rw} + w_{sa}J_{sa} + w_{sep}J_{sep}$$

- **Choice of weighting balance is currently being worked**

- **Want to minimize desats to minimize RCS prop usage.**
  - Introducing RCS firings causes discontinuities (not allowed)
  - Instead minimize RW torque effort and calculate wheel speed
  - Post optimization:
    - When a pair reaches speed limit, reset all RW speeds to 0 and increase desat event count
    - Approximation: all desat maneuvers are equivalent and do not appreciably cause planned orbit deviation

The How

- **Defined:**
  - ✓Inputs
  - ✓Objectives
  - ✓Constraints

So now the “how” ....

EOMs and States

- **Using gradient based optimization (this is a trajectory problem)**
  - Need analytical functions and their derivatives
- **Spacecraft is modeled using continuous analytical functions (sympy)**
  - Dynamics modeled in body frame (not shown)

$$\mathbf{I_{veh}} \begin{bmatrix} \dot{\omega}_{veh,x} \\ \dot{\omega}_{veh,y} \\ \dot{\omega}_{veh,z} \end{bmatrix} = \mathbf{T_{RW}} + \mathbf{T_{SA}} + \mathbf{T_{SEP}} + \mathbf{T_{DIST}}$$

  - Equations of motion = 11
    - 3 from vehicle rotational acceleration
    - 4 from reaction angular velocity
    - 2 from solar array angular velocity
    - 2 from solar array angular acceleration

15 States

$$\boldsymbol{y} = \begin{bmatrix} \omega_{veh,x} & = \cdots \\ \omega_{veh,y} & = \cdots \\ \omega_{veh,z} & = \cdots \\ \omega_{rw,1} & = \cdots \\ \omega_{rw,2} & = \cdots \\ \omega_{rw,3} & = \cdots \\ \omega_{rw,4} & = \cdots \\ \omega_{sada,1} & = \cdots \\ \omega_{sada,2} & = \cdots \\ \theta_{sada,1} & = \cdots \\ \theta_{sada,2} & = \cdots \\ q_{att,1} & = \cdots \\ q_{att,2} & = \cdots \\ q_{att,3} & = \cdots \\ q_{att,4} & = \cdots \end{bmatrix}$$

$$\mathbf{I_{veh}}(\theta_{sada,1}, \theta_{sada,2}) = \mathbf{I_{veh,base}} + \mathbf{I_{SA}}(\theta_{sada,1}, \theta_{sada,2})$$

Quaternion to get
J2K vectors in body frame

The How

- **Sympy Analytical Function Example**
- **Fixed kinematic parameters already numeric**

$$\mathbf{I}_{\mathbf{veh}} \begin{bmatrix} \dot{\omega}_{veh,x} \\ \dot{\omega}_{veh,y} \\ \dot{\omega}_{veh,z} \end{bmatrix} = \mathbf{T}_{\mathbf{RW}} + \mathbf{T}_{\mathbf{SA}} + \mathbf{T}_{\mathbf{SEP}} + \mathbf{T}_{\mathbf{DIST}}$$

```
1 Teq_numeric[0]
```

executed in 766ms, finished 00:22:23 2022-01-17

$$\begin{aligned}
&-\left(\left(12024.1967741\sin^2(\theta_{sada1}(t)) + 9524.1967741\cos^2(\theta_{sada1}(t))\right)\omega_{yveh}(t) + 1250.0\omega_{xveh}(t)\sin(2\theta_{sada1}(t))\right)\omega_{zveh}(t) \\
&-\left(\left(12024.1967741\sin^2(\theta_{sada2}(t)) + 9524.1967741\cos^2(\theta_{sada2}(t))\right)\omega_{yveh}(t) - 1250.0\omega_{xveh}(t)\sin(2\theta_{sada2}(t))\right)\omega_{zveh}(t) \\
&+ 2500.0\left(\omega_{zveh}(t) - 1\right)\omega_{yveh}(t) + 2500.0\left(\omega_{zveh}(t) + 1\right)\omega_{yveh}(t) \\
&+ \left(9524.1967741\sin^2(\theta_{sada1}(t)) + 12024.1967741\cos^2(\theta_{sada1}(t))\right)\frac{d}{dt}\omega_{xveh}(t) \\
&+ \left(9524.1967741\sin^2(\theta_{sada2}(t)) + 12024.1967741\cos^2(\theta_{sada2}(t))\right)\frac{d}{dt}\omega_{xveh}(t) \\
&- \left(-0.484\sin(\theta_{1dsm1}(t))\sin(\theta_{2dsm1}(t)) + 0.769\cos(\theta_{1dsm1}(t)) + 0.02\right)\mathrm{F}_{\mathrm{bht1b}}(t)\sin(\theta_{2dsm1}(t)) \\
&- \left(-0.484\sin(\theta_{1dsm1}(t))\sin(\theta_{2dsm1}(t)) + 1.343\cos(\theta_{1dsm1}(t)) + 0.02\right)\mathrm{F}_{\mathrm{bht1a}}(t)\sin(\theta_{2dsm1}(t)) \\
&- \left(-0.484\sin^2(\theta_{1dsm2}(t)) - 1.125\cos(\theta_{1dsm2}(t)) + 0.02\right)\mathrm{F}_{\mathrm{bht2a}}(t)\sin(\theta_{1dsm2}(t)) \\
&- \left(-0.484\sin^2(\theta_{1dsm2}(t)) - 0.769\cos(\theta_{1dsm2}(t)) + 0.02\right)\mathrm{F}_{\mathrm{bht2b}}(t)\sin(\theta_{1dsm2}(t)) + 1.0414\,\mathrm{F}_{\mathrm{aeps1}}(t)\sin\left(\theta_{1gsm1}(t)\right)\cos\left(\theta_{2gsm1}(t)\right) \\
&+ 0.02\,\mathrm{F}_{\mathrm{aeps1}}(t)\sin\left(\theta_{2gsm1}(t)\right) + 1.0414\,\mathrm{F}_{\mathrm{aeps2}}(t)\sin\left(\theta_{1gsm2}(t)\right)\cos\left(\theta_{2gsm2}(t)\right) - 0.02\,\mathrm{F}_{\mathrm{aeps2}}(t)\sin\left(\theta_{2gsm2}(t)\right) + 0.484\,\mathrm{F}_{\mathrm{bht1a}}(t)\sin \\
&(\theta_{1dsm1}(t))\cos^2(\theta_{2dsm1}(t)) + 0.484\,\mathrm{F}_{\mathrm{bht1b}}(t)\sin(\theta_{1dsm1}(t))\cos^2(\theta_{2dsm1}(t)) + 0.484\,\mathrm{F}_{\mathrm{bht2a}}(t)\sin(\theta_{1dsm2}(t))\cos^2(\theta_{1dsm2}(t)) + 0.484\,\mathrm{F}_{\mathrm{bht2b}} \\
&(t)\sin(\theta_{1dsm2}(t))\cos^2(\theta_{1dsm2}(t)) - 0.39\sqrt{2}\omega_{rw15}(t)\omega_{yveh}(t) + 0.39\omega_{rw15}(t)\omega_{zveh}(t) + 0.39\sqrt{2}\omega_{rw26}(t)\omega_{yveh}(t) - 0.39\omega_{rw26}(t)\omega_{zveh}(t) \\
&+ 0.39\sqrt{2}\omega_{rw37}(t)\omega_{yveh}(t) + 0.39\omega_{rw37}(t)\omega_{zveh}(t) - 0.39\sqrt{2}\omega_{rw48}(t)\omega_{yveh}(t) - 0.39\omega_{rw48}(t)\omega_{zveh}(t) - 5000.0\omega_{xveh}(t)\sin(\theta_{sada1}(t))\cos \\
&(\theta_{sada1}(t))\frac{d}{dt}\theta_{sada1}(t) - 5000.0\omega_{xveh}(t)\sin(\theta_{sada2}(t))\cos(\theta_{sada2}(t))\frac{d}{dt}\theta_{sada2}(t) + 0.78\omega_{yveh}(t)\omega_{zveh}(t) + 2500.0\omega_{yveh}(t)\cos \\
&(2\theta_{sada1}(t))\frac{d}{dt}\theta_{sada1}(t) - 2500.0\omega_{yveh}(t)\cos(2\theta_{sada2}(t))\frac{d}{dt}\theta_{sada2}(t) + 1250.0\sin(2\theta_{sada1}(t))\frac{d}{dt}\omega_{yveh}(t) - 1250.0\sin(2\theta_{sada2}(t))\frac{d}{dt}\omega_{yveh}(t) \\
&- 0.39\frac{d}{dt}\omega_{rw15}(t) - 0.39\frac{d}{dt}\omega_{rw26}(t) - 0.39\frac{d}{dt}\omega_{rw37}(t) - 0.39\frac{d}{dt}\omega_{rw48}(t) - 136926.826\frac{d}{dt}\omega_{xveh}(t) - 183.0\frac{d}{dt}\omega_{yveh}(t) + 353.0\frac{d}{dt}\omega_{zveh}(t)
\end{aligned}$$

The How

- **Vehicle dynamics are formulated in the body frame and utilize roll rate as base state**
- **Need to map sun/earth/thrust vectors into vehicle frame**
- **Direct Collocation solves at discrete nodes, there is no explicit integration to integrate the attitude rate quaternion**
- **Use quaternion derivatives**

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\omega)\mathbf{q}$$

$$\ddot{\mathbf{q}} = \frac{1}{4}\boldsymbol{\Omega}(\omega)\boldsymbol{\Omega}(\omega)\mathbf{q} + \frac{1}{2}\dot{\boldsymbol{\Omega}}(\dot{\omega})\mathbf{q}$$

$$\boldsymbol{\Omega}(\omega) = \begin{bmatrix} 0 & \omega_{veh,z} & -\omega_{veh,y} & \omega_{veh,x} \\ -\omega_{veh,z} & 0 & \omega_{veh,x} & \omega_{veh,y} \\ \omega_{veh,y} & -\omega_{veh,x} & 0 & \omega_{veh,z} \\ -\omega_{veh,x} & -\omega_{veh,y} & -\omega_{veh,z} & 0 \end{bmatrix}$$

- **In a 2nd order finite Taylor series**

$$f(x) = f(x_o) + f'(x_o)(x - x_o) + \frac{1}{2}f''(x_o)(x - x_o)$$

$$\dot{\mathbf{q}}_{\mathbf{update}} = \frac{\mathbf{q_t} - \mathbf{q_{t-1}}}{\Delta t} = \frac{1}{2}\boldsymbol{\Omega}(\omega_\mathbf{t})\mathbf{q_t} - \frac{1}{8}\boldsymbol{\Omega}(\omega_\mathbf{t})\boldsymbol{\Omega}(\omega_\mathbf{t})\mathbf{q_t}\Delta t - \frac{1}{4}\dot{\boldsymbol{\Omega}}(\dot{\omega})\mathbf{q_t}\Delta t$$

- **Use only the 3 vector entries from** $\dot{\mathbf{q}}_{\mathbf{update}}$. **The scaler comes from a "slack" trajectory**
- **Then add a unity constraint at every node:** $q_{att,1}^2 + q_{att,2}^2 + q_{att,3}^2 + q_{att,4}^2 - 1 = 0$

The How

- **Work Flow**
  - Create model equations of motion, constraints equations, and objective in sympy
  - Create constraint equations
    - This includes making sure that the state the optimizer choses for node n ... leads to the states it chooses at node n+1
  - Transcribing these equations into the direct collocation framework
    - This can be tedious, so using opty package
      - Create constraints (including dynamic consistency from EOMs)
      - Generates symbolic constraints jacobian
      - Creates C++ compiled library
      - Optimizes using IPOPT (via cyipopt)

Results

- **Preliminary Results**
  - Preliminary qualitative findings:
    - Solution very sensitive to objective weighting
    - There seems to be a preference for using the solar array momentum (AMM)
      - Most pronounced at perigee passage
    - Optimizing over full 1-year EOR is unneeded and even undesired
      - Produces strategies that reach law of averages
      - 7-day mission planning window produces better results and reflects typical ops
  - Currently investigating moving thrust alignment and solar incidence from objective function and into constraint functions
    - Zero or single Pareto front
      - Hard limit thrust misalignment and trade solar misalignment vs. momentum usage
      - Hard limit solar misalignment and trade thrust misalignment vs. momentum usage

Results and Work Ahead

- **Tools:**
  - sympy:https://www.sympy.org/en/index.html
  - opty: https://readthedocs.org/projects/opty/
  - cyipopt: https://cyipopt.readthedocs.io/en/stable/

NASA